BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

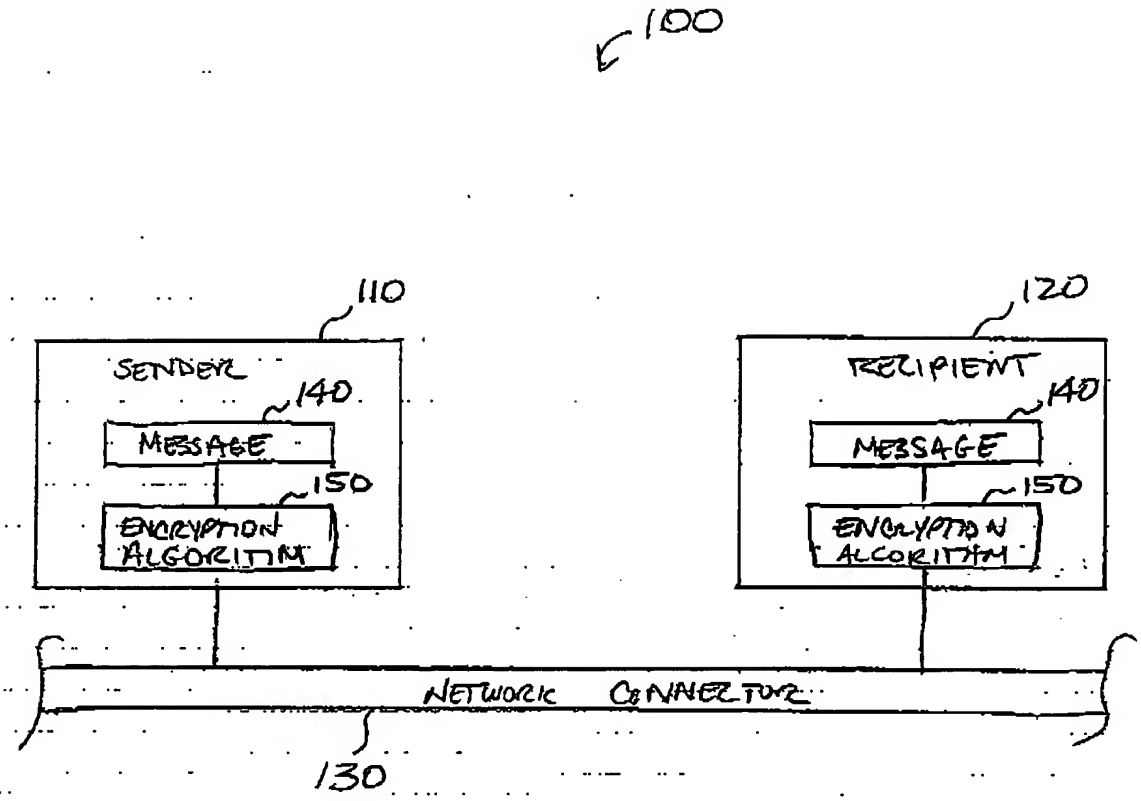

FIG. 1A (PRIOR ART)

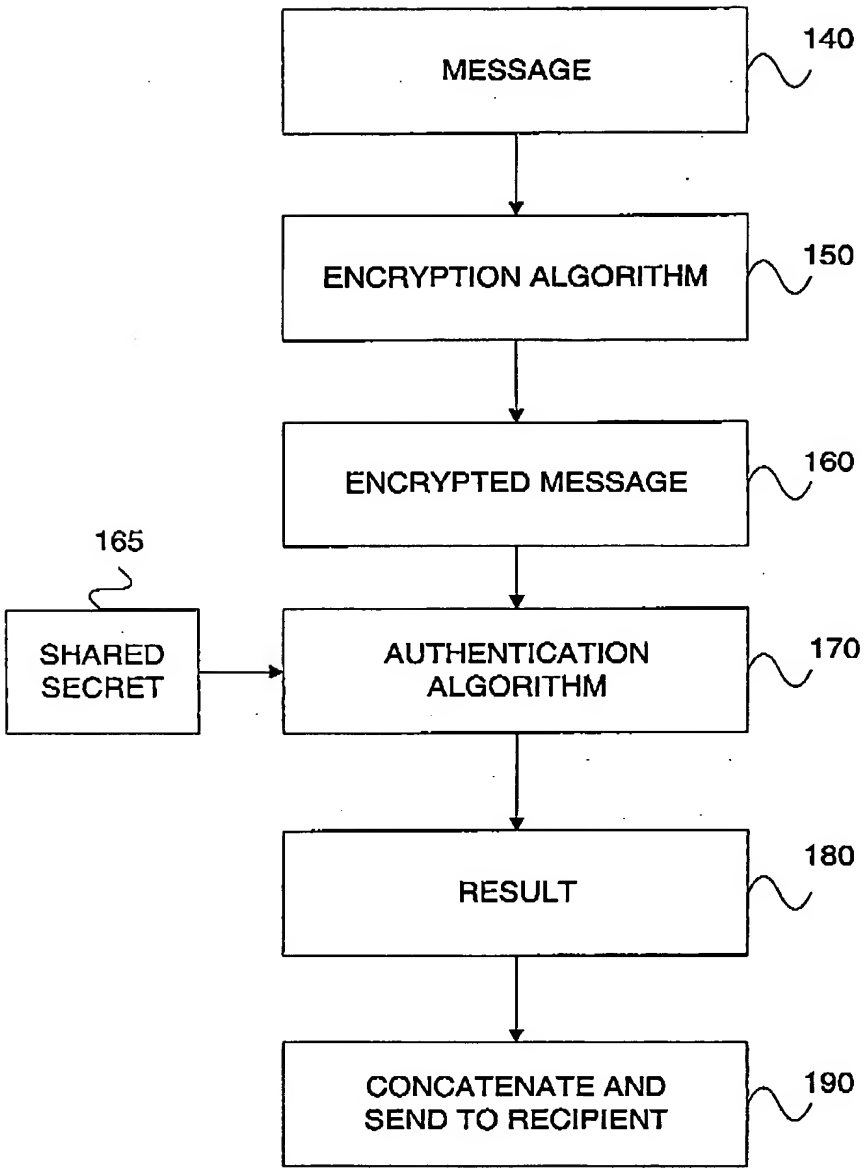
MESSAGE
140
ENCRYPTION ALGORITHM
150
ENCRYPTED MESSAGE
160
165
SHARED SECRET
AUTHENTICATION ALGORITHM
170
RESULT
180
CONCATENATE AND SEND TO RECIPIENT
190

FIG. 1B

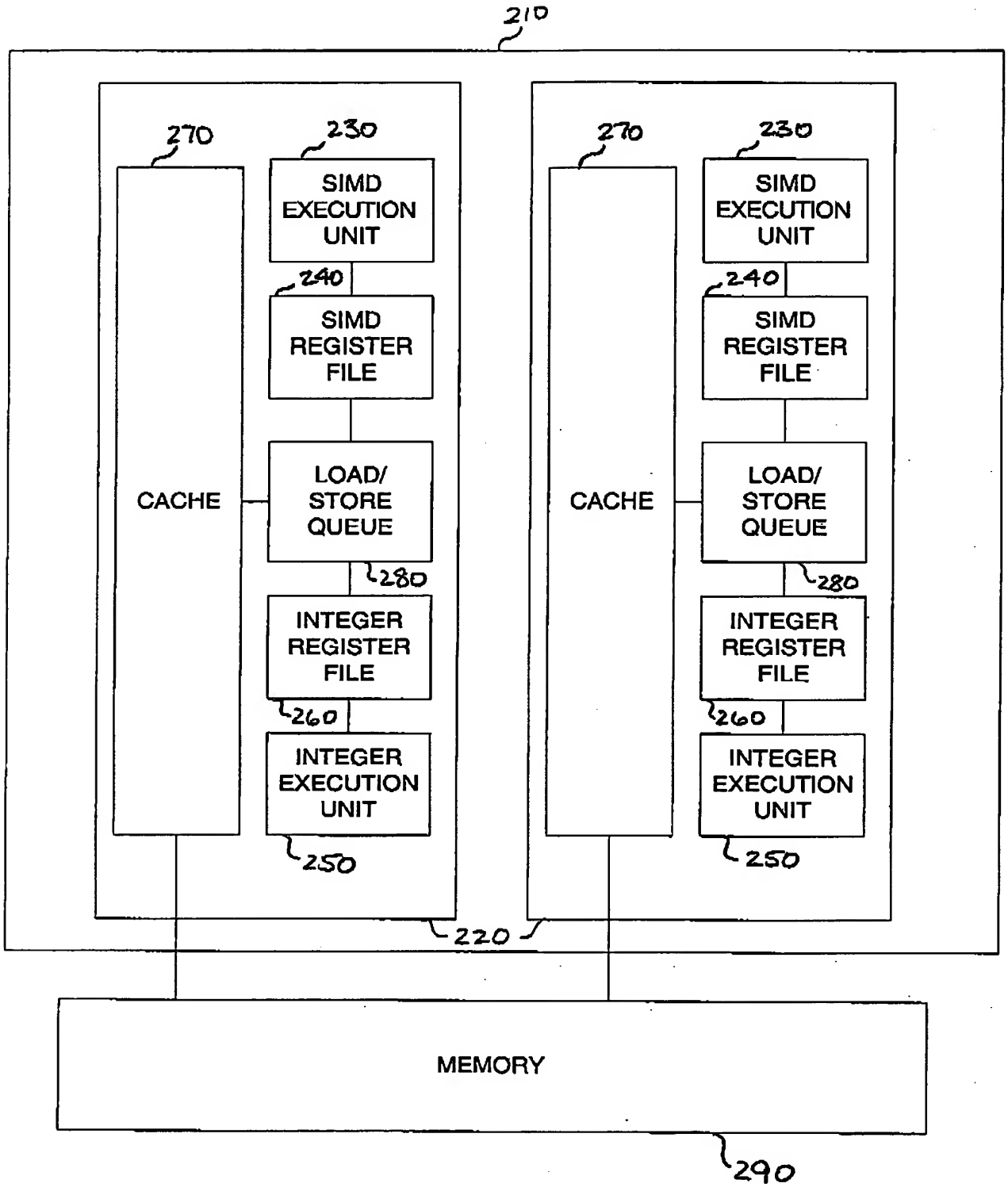
210
270
230
SIMD EXECUTION UNIT
240
SIMD REGISTER FILE
CACHE
LOAD/ STORE QUEUE
280
INTEGER REGISTER FILE
260
INTEGER EXECUTION UNIT
250
220
MEMORY
290

FIG. 2

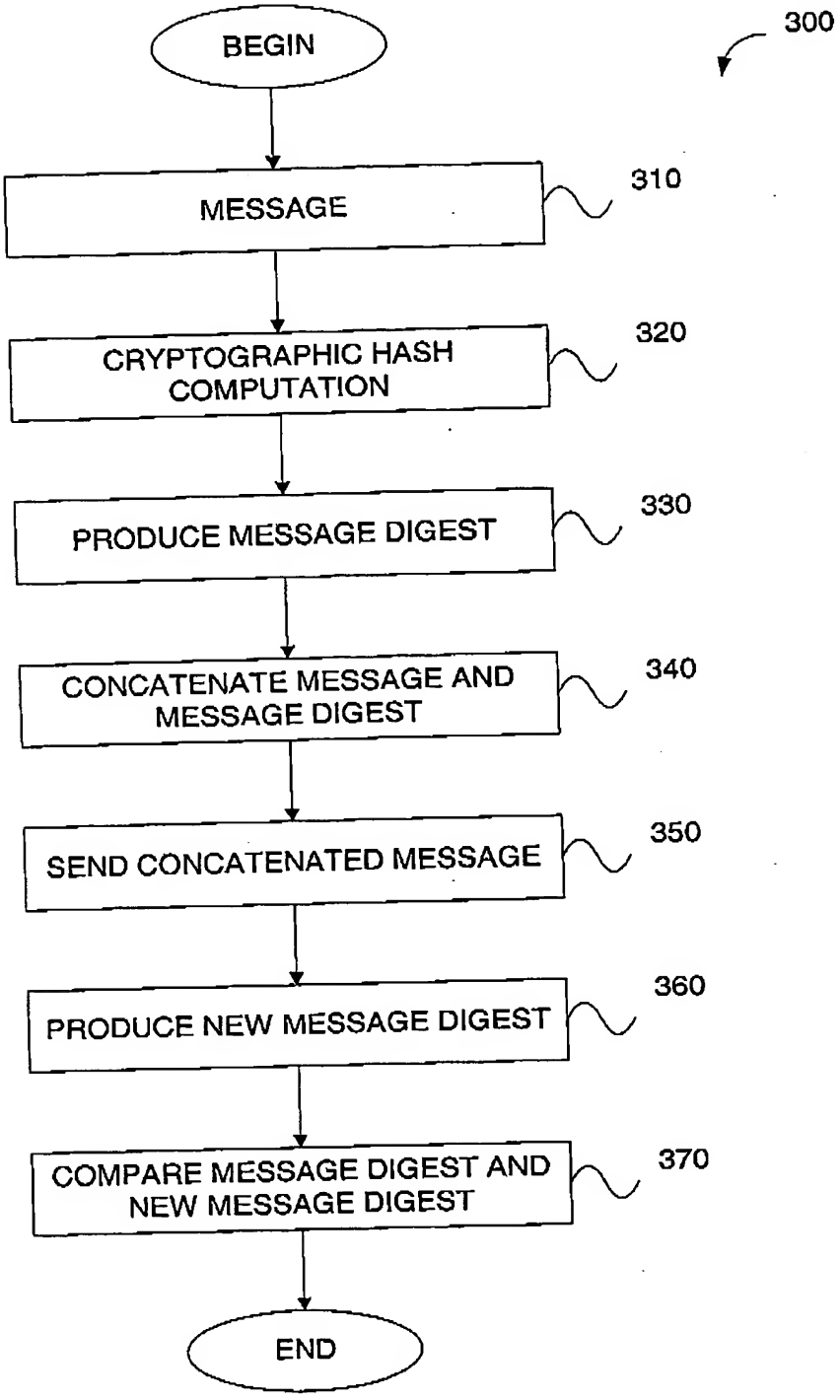
300
BEGIN
MESSAGE
310
CRYPTOGRAPHIC HASH COMPUTATION
320
PRODUCE MESSAGE DIGEST
330
CONCATENATE MESSAGE AND MESSAGE DIGEST
340
SEND CONCATENATED MESSAGE
350
PRODUCE NEW MESSAGE DIGEST
360
COMPARE MESSAGE DIGEST AND NEW MESSAGE DIGEST
370
END

FIG. 3

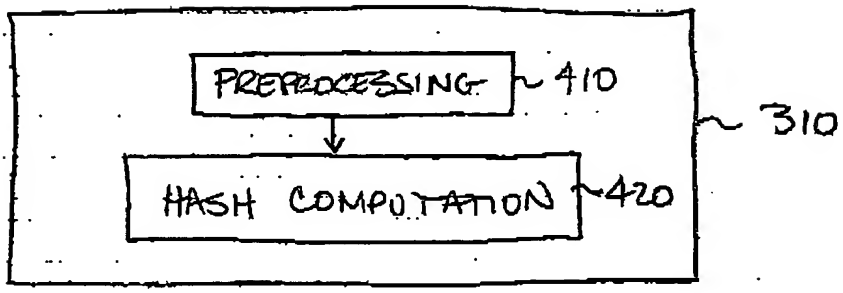
PREPROCESSING
410
HASH COMPUTATION
420
310

FIG. 4

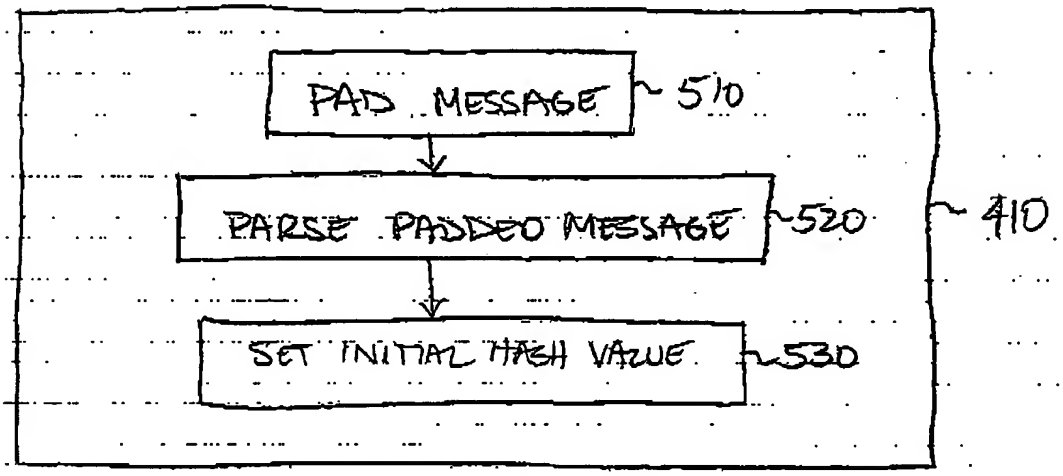
PAD MESSAGE
510
PARSE PADDED MESSAGE
520
SET INITIAL HASH VALUE
530
410

FIG. 5

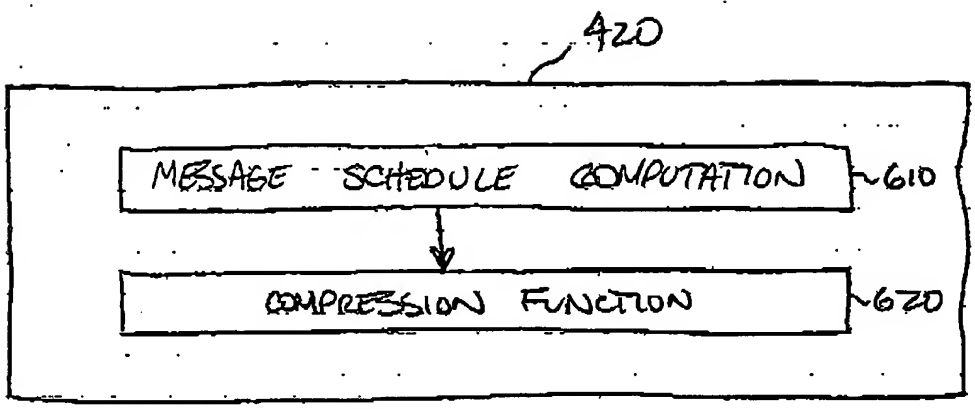
420
MESSAGE SCHEDULE COMPUTATION
610
COMPRESSION FUNCTION
620

FIG. 6

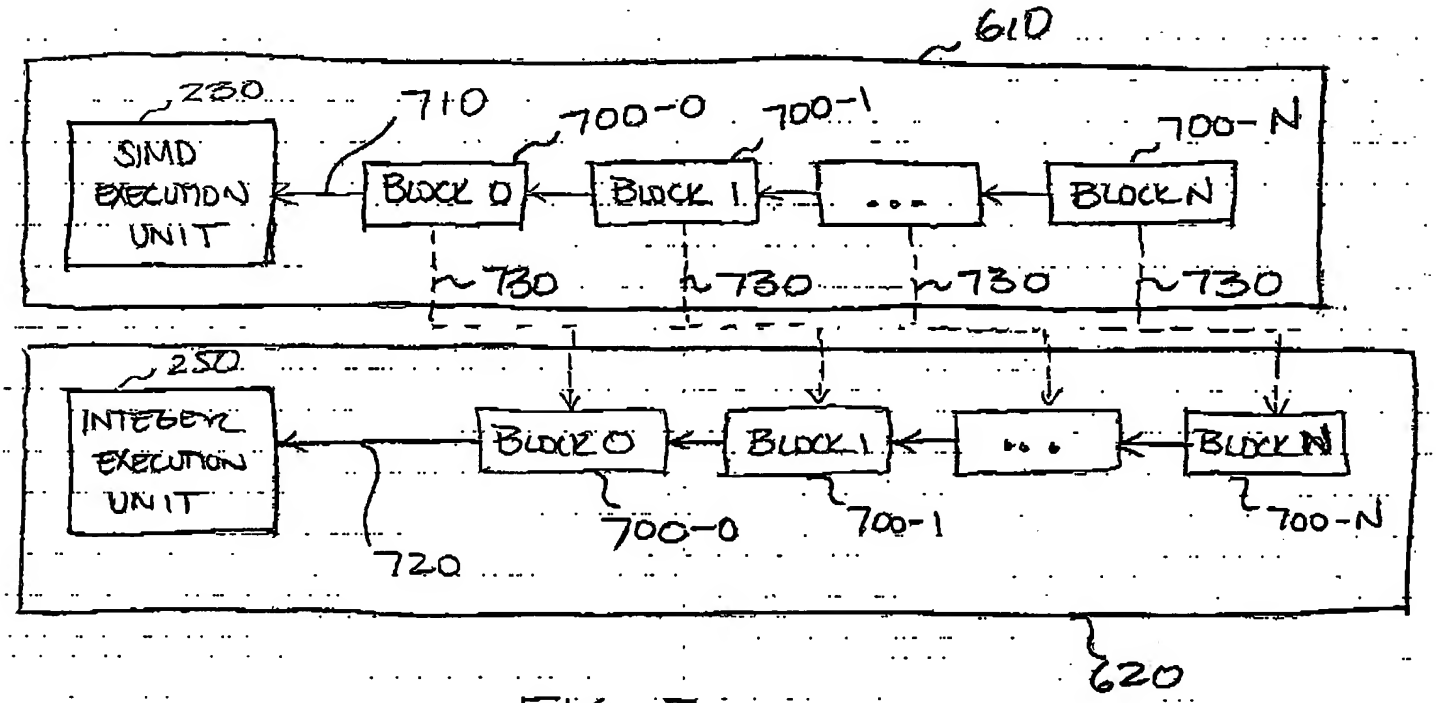
610
230
710
700-0
700-1
700-N
SIMD EXECUTION UNIT
BLOCK 0
BLOCK 1
...
BLOCK N
730
730
730
730
250
INTEGER EXECUTION UNIT
BLOCK 0
BLOCK 1
...
BLOCK N
700-0
700-1
700-N
720
620

FIG. 7

800

```
Wj = Mj for j = 0 to 15
for j = 16 to 79
{
   Wj = Rotl1(Wj-3 ⊕ Wj-8 ⊕ Wj-14 ⊕ Wj-16)
}
```

FIG. 8A

850

```
for j = 0 to 79
{

  T = rotl5(a) + fj(b,c,d) + e + kj + wj
  e = d
  d = c
  c = rotl30(b)
  b = a
  a = T
}

where:

  fj(x, y, z)  = (x&y) ⊕ (~x&z)          for j =  0 to 19
               = x ⊕ y ⊕ z               for j = 20 to 39
               =     (x&y)⊕(x&z)⊕(y&z)   for j = 40 to 59
          = x ⊕ y ⊕ z                  for j = 60 to 79

kj = 0x5a827999                        for j =  0 to 19
   = 0x6ed9eba1                        for j = 19 to 39
   = 0x8f1bbcdc                        for j = 40 to 59
   = 0xca62c1d6                        for j = 60 to 79
```

FIG. 8B

900

```
Wj = Mj for for j = 0 to 15
for j = 16 to 63
{
   Wj = S1 (Wj-2) + Wj-7 + S0 (Wj-15) + Wj-16
}

where:

S0(x) = Rotr7(x) ^ Rotr18(x) ^    Shr3(x)
S1(x) = Rotr17(x) ^ Rotr19(x) ^    Shr10(x)
```

FIG. 9A

950

```
for j = 0 to 63
{
  T1 = h + sig1(e) + ch(e,f,g) + kj + Wj
  T2 = sig0(a) + maj(a,b,c)
  h = g
  g = f
  f = e
  e = d + T1
  d = c
  c = b
  b = a
  a = T1 + T2
}

where:

sig0(e) = rotr2(e) ⊕ rotr13(e) ⊕ rotr22(e)
sig1(a) = rotr6(a) ⊕ rotr11(a) ⊕ rotr25(a)
ch(e,f,g) = (e&f) ⊕ (~e&g)
maj(a,b,c) = (a&b) ⊕ (a&c) ⊕ (b&c)
```

FIG. 9B

1000

```
Wj = mj for j = 0 to 15
for j = 16 to 79
{
   Wj = gamma1(Wj-2) + Wj-7 + gamma0(tj-15) + Wj-16
}

where:

gamma0(x) = rotr1(x) ⊕ rotr8(x) ⊕ shr7(x)
gamma1(x) = rotr19(x) ⊕ rotr61(x) ⊕ shr6(x)
```

FIG. 10A

1050

```
for j = 0 to 79
{
  T1 = h + sig1(e) + ch(e,f,g) + kj + wj
  T2 = sig0(a) + maj(a,b,c)
  h = g
  g = f
  f = e
  e = d + T1
  d = c
  c = b
  b = a
  a = T1 + T2
}

where:

sig0(e) = rotr28(e) ⊕ rotr34(e) ⊕ rotr39(e)
sig1(a) = rotr14(a) ⊕ rotr18(a) ⊕ rotr41(a)
ch(e,f,g) = (e&f) ⊕ (~e&g)
maj(a,b,c) = (a&b) ⊕ (a&c) ⊕ (b&c)
```

FIG. 10B